Converting Flux Files to dk2nu Format

Robert Hatcher

Fermilab Computing Division

Beam Simulation Mtg 2012-12-14

Access to the code

- Setup FNAL UPS version:

```
source /nusoft/app/alt/setup.sh
setup genie v3665 -q e2:debug  # get GENIE + ROOT + gcc setup
setup dk2nu test4 -q e2:debug  # new version 2012-12-14 (to make this job easier)
```

- Create working area; copy/modify the locations file:

```
cd /nova/app/users/$USER/mydir   # your own working area (needs to be "app")
cp  $DK2NU/etc/locations.txt mylocations.txt
ln -s $DK2NU/convert/flugg/convert_flugg.C .          # for convenience
ln -s $DK2NU/convert/g4minerva/convert_g4minerva.C .  # for convenience
# edit mylocations.txt
```

- add/adjust points as desired
  - any number of locations (test small changes in NOvA positions?)
- must be in beam coordinates and units (cm)

Converting the ntuple

- Different for each “flavor” (flugg, g4minerva, etc)

```
root -b -q $DK2NU/snippets/load_dk2nu.C \
‘convert_flugg.C+(“myflugg.root”,42,”./mylocations.txt”,”none”,false)’
```

```
root -b -q $DK2NU/snippets/load_dk2nu.C \
‘convert_g4minerva.C+(“myg4.root”,42,”./mylocations.txt”,”none”,false)’
```

- these commands assume you’ve made the symbolic links on the previous page
- cut-and-pasting above lines might get the single & double quotes wrong (check that they are both the character one gets by typing the key next to the return key); you might need to remove the “\” continuation line character as well.
- resulting files: `myflugg_to_dk2nu.root & myg4_to_dk2nu.root`
- 42 = job # (original files don’t always have non-zero “run”)
- “none” turns off the cross-checks (otherwise use “NOVA” for flugg)
- false = turn of print of last entry in result ntuple

Interactive Plotting

- Plot the resulting files

```
root $DK2NU/snippets/load_dk2nu.C   myflugg_to_dk2nu.root
```

~~dkmetaTree->Scan("location.name","","colsize=25");~~

doesn't show all entries due to root looping bug (Savanna report 98899)

```
TCanvas* c1 = new TCanvas();   c1->SetLogy();

dk2nuTree->Draw("nuray[1].E","nimpwt*nuray[1].wgt"); // minos near flux

dk2nuTree->SetLineColor(kRed);

dk2nuTree->Draw("nuray[3].E","nimpwt*nuray[3].wgt","SAME");  // nova near flux
```

- where [1] and [3] pick out particular locations
  - this is just an example of using the multiple locations in the same file
- filling histogram w/ E_ν with right weight (position*importance)
  - you might want to also select neutrino flavor, e.g.
    - "nimpwt*nuray[3].wgt*(ntype==14)"
- you'll need to plot the two files and then overlay the plots

Multiple (similar) files

- for multiple files of the same type (flugg vs. g4minerva)

```
root $DK2NU/snippets/load_dk2nu.C
TChain* flugg = new TChain("dk2nuTree");
flugg->Add("*flugg*_to_dk2nu.root");
flugg->Draw("nuray[3].E","nimpwt*nuray[3].wgt*(ntype==-14)","");
TChain* g4minerva = new TChain("dk2nuTree");
g4minerva->SetLineColor(kRed);
g4minerva->Add("*g4*_to_dk2nu.root");
g4minerva->Draw("nuray[3].E","nimpwt*nuray[3].wgt*(ntype==-14)","SAME");
```

  - don’t want to mix flugg and g4minerva files in same chain; but one can have 2
  - if two collections represent different numbers of POTs you’ll need a scale factor to have a sensible overlay plot
  - histogram max set by first plot; reverse order if g4minerva > flugg